IN THE SPECIFICATION:

The specification as amended below with replacement paragraphs shows added text with underlining and deleted text with ~~strikethrough~~.

Please REPLACE paragraph [0006] of the Substitute Specification with the following paragraph:

[0006] With the inventive solution, contexts are introduced for forming a plurality of indirection ~~stages~~ levels for managing the identifiers. This provides efficient methods for repairing "broken links" without introducing global, central management functions.

- **No central management**: Management is through the context hierarchy. This means that each context contains all the necessary information.
- **Support for distributed work**: The context hierarchy can be broken down and reassembled as desired. This means that projects can be branched and merged without difficulty.
- **Little modification effort**: The context hierarchy makes it immediately clear where changes to identifiers are to be reconstructed. The changes also need to be made only on the context objects in question.

Please REPLACE paragraph [0012] of the Substitute Specification with the following paragraph:

[0012] This now poses the problem of how consistency can be produced as far as possible without demanding excessive effort from the developer of ES-Autos or OVA tools for this purpose. These mechanisms should also not be transparent to the parts of the API which are used by the OVA tools. A client application should always be occupied with the ES-Auto names, for example, and not with cryptic IDs (see FIGURE 1).

- ***Problematical actions***

It is first necessary to examine the actions which conceal potential risks. These are, firstly, all unidirectional relationships, and, secondly, actions which "pass by" the data model servers.

- Unidirectional links

Unidirectional links are problematical because it is not possible to tell from an action that an inconsistency is being produced. If a link is used to point to a file in a Word document, for example, and this file is renamed at a later point in time, the Word document is not told anything

about this and will not find the file again. This problem can be eliminated only by a central authority which knows where the file can be found.

- "Dumb" actions

In this regard, dumb actions denote actions carried out without the knowledge of the data model. An example is renaming an object using the IStorage interface (IStorage::RenameElement). Such actions are always possible for standardized data access. In this case too, a central authority could help to minimize the problem. An important aspect in both cases is, above all, error recognition, and if possible also error elimination.

- ***Object ID moniker***

As described above, a central office can manage the objects such that they are (virtually) uniquely identifiable. All the objects are therefore referenced using object IDs which can be resolved by the central office~~, in our case the Active Directory Service~~. This ID is independent of all actions; it is allocated upon creation of the object and then does not change again so long as no other object having the same ID exists. This will only occur in the case of copying outside the data model, however. Each container allocates a name when an embedded object is created.

Please REPLACE paragraph [0035] of the Substitute Specification with the following paragraph:

[0035] In summary, the invention relates to a system and method for object identification in distributed hierarchical systems, in particular in automation systems. To ensure object identification for operations such as moving, copying, renaming, etc., the invention proposes introducing contexts for forming a plurality of indirection ~~stages~~ levels for managing identifiers. This provides efficient methods for repairing "broken links" without introducing global, central management functions.